1

WRDC-TR-90-8007
Volume V
Part 38

INTEGRATED INFORMATION SUPPORT SYSTEM (IISS)
Volume V - Common Data Model Subsystem
Part 38 - DDL to NDDL Translator Build Instructions

M. Apicella, S. Singh

Control Data Corporation
Integration Technology Services
2970 Presidential Drive
Fairborn, OH 45324-6209

September 1990

Final Report for Period 1 April 1987 - 31 December 1990

Approved for Public Release; Distribution is Unlimited

MANUFACTURING TECHNOLOGY DIRECTORATE
WRIGHT RESEARCH AND DEVELOPMENT CENTER
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433-6533

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever, regardless whether or not the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data. It should not, therefore, be construed or implied by any person, persons, or organization that the Government is licensing or conveying any rights or permission to manufacture, use, or market any patented invention that may in any way be related thereto.

This technical report has been reviewed and is approved for publication.

This report is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations

David Judson

DAVID L. JUDSON, Project Manager
WRDC/MTI
Wright-Patterson AFB, OH 45433-6533

25 July 91
DATE

FOR THE COMMANDER:

Bruce Rasmussen

BRUCE A. RASMUSSEN, Chief
WRDC/MTI
Wright-Patterson AFB, OH 45433-6533

25 July 91
DATE

If your address has changed, if you wish to be removed form our mailing list, or if the addressee is no longer employed by your organization please notify WRDC/MTI, Wright-Patterson Air Force Base, OH 45433-6533 to help us maintain a current mailing list.

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION
Unclassified

1b. RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION/DOWNGRADING SCHEDULE

3. DISTRIBUTION/AVAILABILITY OF REPORT
Approved for Public Release;
Distribution is Unlimited.

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
UM 620341411

5. MONITORING ORGANIZATION REPORT NUMBER(S)
WRDC-TR-90-8007 Vol. V, Part 38

6a. NAME OF PERFORMING ORGANIZATION
Control Data Corporation;
Integration Technology Services

6b. OFFICE SYMBOL *(if applicable)*

7a. NAME OF MONITORING ORGANIZATION
WRDC/MTI

6c. ADDRESS *(City,State, and ZIP Code)*
2790 Presidential Drive
Fairborn, OH 45324-6209

7b. ADDRESS *(City, State, and ZIP Code)*
WPAFB, OH 45433-6533

8a. NAME OF FUNDING/SPONSORING ORGANIZATION
Wright Research and Development Center,
Air Force Systems Command, USAF

8b. OFFICE SYMBOL *(if applicable)*
WRDC/MTI

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUM.
F33600-87-C-0464

8c. ADDRESS *(City, State, and ZIP Code)*
Wright-Patterson AFB, Ohio 45433-6533

10. SOURCE OF FUNDING NOS.

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO.
78011F	595600	F95600	20950607

11. TITLE *(Include Security Classification)*
DΓ See block 19

12. PERSONAL AUTHOR(S)
Control Data Corporation: Apicella, M. L., Singh, S.

13a. TYPE OF REPORT
Final Report

13b. TIME COVERED
4/1/87-12/30/90

14. DATE OF REPORT *(Yr.,Mo.,Day)*
1990 September 30

15. PAGE COUNT
15

16. SUPPLEMENTARY NOTATION
WRDC/MTI Project Priority 6203

17. COSATI CODES

FIELD	GROUP	SUB GR.
1308	0905	

18. SUBJECT TERMS *(Continue on reverse if necessary and identify block no.)*

19. ABSTRACT *(Continue on reverse if necessary and identify block number)*

This document describes how to construct the DDL to NDDL Translators. It also lists the prerequisites for the DDL to NDDL Translators and provides step-by-step instructions for building the DB2 and Total translators.

BLOCK 11:

INTEGRATED INFORMATION SUPPORT SYSTEM
Vol V - Common Data Model Subsystem

Part 38 - DDL to NDDL Translator Build Instructions

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT
UNCLASSIFIED/UNLIMITED x SAME AS RPT. DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION
Unclassified

22a. NAME OF RESPONSIBLE INDIVIDUAL
David L. Judson

22b. TELEPHONE NO. *(Include Area Code)*
(513) 255-7371

22c. OFFICE SYMBOL
WRDC/MTI

EDITION OF 1 JAN 73 IS OBSOLETE

FOREWORD

This technical report covers work performed under Air Force Contract F33600-87-C-0464, DAPro Project. This contract is sponsored by the Manufacturing Technology Directorate, Air Force Systems Command, Wright-Patterson Air Force Base, Ohio. It was administered under the technical direction of Mr. Bruce A. Rasmussen, Branch Chief, Integration Technology Division, Manufacturing Technology Directorate, through Mr. David L. Judson, Project Manager. The Prime Contractor was Integration Technology Services, Software Programs Division, of the Control Data Corporation, Dayton, Ohio, under the direction of Mr. W. A. Osborne. The DAPro Project Manager for Control Data Corporation was Mr. Jimmy P. Maxwell.

The DAPro project was created to continue the development, test, and demonstration of the Integrated Information Support System (IISS). The IISS technology work comprises enhancements to IISS software and the establishment and operation of IISS test bed hardware and communications for developers and users.

The following list names the Control Data Corporation subcontractors and their contributing activities:

SUBCONTRACTOR	ROLE
Control Data Corporation	Responsible for the overall Common Data Model design development and implementation, IISS integration and test, and technology transfer of IISS.
D. Appleton Company	Responsible for providing software information services for the Common Data Model and IDEF1X integration methodology.
ONTEK	Responsible for defining and testing a representative integrated system base in Artificial Intelligence techniques to establish fitness for use.

Simpact Corporation	Responsible for Communication development.
Structural Dynamics Research Corporation	Responsible for User Interfaces, Virtual Terminal Interface,and Network Transaction Manager design, development, implementation, and support.
Arizona State University	Responsible for test bed operations and support.

TABLE OF CONTENTS

SECTION 1

INTRODUCTION

To construct the DDL to NDDL TRANSLATORS, the following steps must be executed.

- Precompile the program with embedded NDML
- Generate the Translators Request Processor Main Program
- Compile and insert into the object library (GENOLB) the generated Translators Request Processor Main Program
- Create the DB2 and Total Translator Executables

Section 2 lists the prerequisites of the DDL to NDDL Translators. Section 3 contains the step by step instructions for building the DB2 and Total Translators.

SECTION 2

PREREQUISITES

The Prerequisites to creating the DDL to NDDL TRANSLATORS environment are:

1. Existence of an object library DDLTOLB in the directory cdmdir:[tools.ddlt] for the Impact software.

2. Existence of an object library GENOLB in the directory cdmdir:[tools.ddlt] for the generated code.

3. Existence of a a FORMS directory; this is the directory pointed to by the logical IISSULIB.

4. All the software must be compiled and placed in DDLTOLB according to normal Integration and Testing procedures.

5. The NDDL and NDML executables must be available.

SECTION 3

PROCEDURES TO BUILD THE TRANSLATOR EXECUTABLES

The following steps must be executed to construct the DDL to NDDL Translator Executables:

1. Precompile and compile the NDML-embedded source code. Proceed as follows:

   ```
   $@BLDTRAN
   PRECOMPILE AND COMPILE A GROUP OF PRC'S
   ------------------------------
   NAME OF THE APPLICATION>: TRANSL

   NAME OF THE PRC FILE (C/R TO STOP, LEAVE, .PRC OFF): TVERDT
   NAME OF THE PRC FILE (C/R TO STOP, LEAVE, .PRC OFF): <CR>

   NDML PRECOMPILE SUCCESSFULLY COMPLETED
   BEGIN COMPILING GENERATED CODE
   RESULTS OF COMPILE CAN BE FOUND ON TRANSL.MSG
   ```

2. Execute the procedure file LNKTRAN.COM to generate the RP-MAIN, compile the RP-MAIN, and place in GENOLB, generate the opt files and link the applications. The two executables that will be created are TOTTRANZ.EXE and D2TRANZ.EXE. The NTM tables are updated automatically. Proceed as follows:

   ```
   $ @LNKTRAN
   ```

3. Flan the file TRANS.FDL. The form definition files will be:

   TRANS.FD

4. Define the Translator Functions using the User Interface utility SYSGEN. The username, password and role are MORENC, STANLEY and MANAGER respectively. The DB2 function is DB2TRANZ and the DB2 application is GRDB2TRANZ. The TOTAL function is TOTTRANZ and the TOTAL application is GRTOTTRANZ.

   The following pages contain listings for:

   BLDTRAN.COM
   LNKTRAN.COM

```
$!
$!
$!    BLDTRAN.COM
$!
$!
$WS:= WRITE SYS$OUTPUT
$DEFINE IISSGLIB "CDMDIR:[TOOLS.DDLT]GENOLB.OLB"
$DEFINE CDMTEMPS "CDMDIR:[TOOLS.DDLT.TEMPS]"
$WS "PRECOMPILE AND COMPILE A GROUP OF PRC'S"
$WS "-----------------------------"
$!
$INQUIRE AP " NAME OF THE APPLICATION>"
$ CREATE 'AP'.DAT
$ OPEN/WRITE   NDMLIN  'AP'.IN
$NEXT:
$ INQUIRE PRC   "NAME OF PRC FILE (C/R TO STOP, LEAVE .PRC OFF)"
$ IF PRC .EQS. "" THEN GOTO INDONE
$ APPEND 'PRC'.PRC 'AP'.DAT
$ GOTO NEXT
$ INDONE:
$!
$!  5/24/88:  FDL stuff added in because NDML now requires fixed
$!            length .DAT files since conversion to FIOPS.
$!
$ OPEN/WRITE FDLIN CDMDIR:[COM]FIX.FDL
$ WRITE FDLIN "IDENT   ""23-FEB-1988 09:49:43   VAX-11 FDL Editor"""
$ WRITE FDLIN "   "
$ WRITE FDLIN "SYSTEM"
$ WRITE FDLIN "        SOURCE                  VAX/VMS"
$ WRITE FDLIN "     "
$ WRITE FDLIN "FILE"
$ WRITE FDLIN "        ALLOCATION              391"
$ WRITE FDLIN "        BEST_TRY_CONTIGUOUS     yes"
$ WRITE FDLIN "        EXTENSION               39"
$ WRITE FDLIN "        ORGANIZATION            sequential"
$ WRITE FDLIN "     "
$ WRITE FDLIN "RECORD"
$ WRITE FDLIN "        BLOCK_SPAN              yes"
$ WRITE FDLIN "        CARRIAGE_CONTROL        carriage_return"
$ WRITE FDLIN "        FORMAT                  fixed"
$ WRITE FDLIN "        SIZE                    80"
$ CLOSE FDLIN
$ CONVERT/PAD=%040/FDL=CDMDIR:[COM]FIX.FDL 'AP'.DAT 'AP'.DAT
$ WRITE NDMLIN "CDMTRAN VAX VAX COBOL NDML COBOL"," ",AP,".DAT ",AP,".ER
$ CLOSE NDMLIN
$ OPEN/WRITE EDIPRC CDMDIR:[COM]EDIT.PROC
$ WRITE EDIPRC "S/80/200/WH"
$ WRITE EDIPRC "EXIT"
$ CLOSE EDIPRC
$ EDIT/COMMAND=CDMDIR:[COM]EDIT.PROC CDMDIR:[COM]FIX.FDL
$ DEASSIGN SYS$OUTPUT
$ CONVERT/PAD=%040/FDL=CDMDIR:[COM]FIX.FDL 'AP'.IN 'AP'.IN
$ DELETE CDMDIR:[COM]FIX.FDL;*, CDMDIR:[COM]EDIT.PROC;*
$!
$!  INPUTS TO PRECOMPILER ARE NOW SET UP
$!   GO AHEAD AND RUN IT:
$!
$ ASSIGN/USER_MODE SYS$COMMAND SYS$INPUT
$ RENAME 'AP'.IN NDML.DAT
$ ASSIGN 'AP'.OUT SYS$OUTPUT
$ RUNNDML
$ delete NDML.DAT;*
$!
$ ALLDONE:
$ DEASSIGN SYS$OUTPUT
$!
$!  check the .out file for errors in precompiling
$!
$OPEN/READ EFLE 'AP'.OUT
$ZR:="0"
$  NERRLOOP:
$   READ/END_OF_FILE=COMPERR EFLE EREC
$     LENG = 'F$LENGTH(EREC)'
$     UN = 'F$LOCATE("UNSUCC",EREC)'
$     IF 'UN' .EQS. 'LENG' THEN GOTO NERRLOOP
$     UN1 = 'UN' - 13
$     UN2 = 'F$EXTRACT(UN1,1,EREC)'
$     IF UN2 .EQS. ZR THEN GOTO NDMLGOOD
$WS "THE PRECOMPILE OF ''AP' HAS ''UN2'  UNSUCCESSFUL ROUTINES"
$WS "CHECK THE ''AP'.ERR FILE FOR ERRORS"
$GOTO EXIT
$COMPERR:
$WS "PRECOMPILE FAILED"
$GOTO EXIT
$!
$!  the precompile was successful, compile the code
$!
$  NDMLGOOD:
$WS "     "
$WS "NDML PRECOMPILE SUCCESSFULLY COMPLETED"
$WS "BEGIN COMPILING GENERATED CODE"
$  DELETE 'AP'.DAT;
$  NDMLGDRD:
$ READ/END_OF_FILE = COMPERR EFLE EREC
$  LENG = 'F$LENGTH(EREC)'
$  UN = 'F$LOCATE("==> USE", EREC)'
$  IF 'UN' .EQS. 'LENG' THEN GOTO NDMLGDRD
$!
$ASSIGN 'AP'.MSG SYS$OUTPUT
$UN1 = 'UN' + 8
$NNAM:='F$EXTRACT(UN1,30,EREC)'
$CLOSE EFLE
$ @'NNAM'
$DEASSIGN SYS$OUTPUT
$WS "RESULTS OF COMPILE CAN BE FOUND ON ''AP'.MSG"
$EXIT:
$DEFINE IISSGLIB "CDMDIR:[TEST]GENOLB.OLB"
```

```
$!
$!        LNKTRAN.COM
$!        THIS USES ORACLE VERSION 5.1
$!
$!
$DEFINE/NOLOG TOOLOLB   CDMDIR:[TOOLS.DDLT]DDLTOLB
$DEFINE/NOLOG IISSGLIB "CDMDIR:[TOOLS.DDLT]GENOLB.OLB"
$DEFINE/NOLOG CDMTEMPS CDMDIR:[TOOLS.DDLT.TEMPS]
$if p1 .eqs. "N" then goto linkTRAN
$WS="WRITE SYS$OUTPUT"
$  LINK:
$DEASSIGN SYS$OUTPUT
$WS " "
$WS "Beginning Generation of Rp-Main"
$LUW="CDMTRAN"
$CDM="CDM/CDM"
$!
$! generate the rp-main
$! NOTE:  this is done automatically if you link
$!
$!
$!  set up .dat file to send to genrpd
$!
$OPEN/WRITE GENRPD.DAT GENRPD.DAT
$WRITE GENRPD.DAT LUW, " ", CDM, " VAX"
$ CLOSE GENRPD.DAT
$ OPEN/WRITE FDLIN FIX.FDL
$ WRITE FDLIN "IDENT   ""23-FEB-1988 09:49:43   VAX-11 FDL Editor"""
$ WRITE FDLIN "   "
$ WRITE FDLIN "SYSTEM"
$ WRITE FDLIN "        SOURCE                  VAX/VMS"
$ WRITE FDLIN "    "
$ WRITE FDLIN "FILE"
$ WRITE FDLIN "        ALLOCATION              3"
$ WRITE FDLIN "        BEST_TRY_CONTIGUOUS     yes"
$ WRITE FDLIN "        EXTENSION               39"
$ WRITE FDLIN "        ORGANIZATION            sequential"
$ WRITE FDLIN "    "
$ WRITE FDLIN "RECORD"
$ WRITE FDLIN "        BLOCK_SPAN              yes"
$ WRITE FDLIN "        CARRIAGE_CONTROL        carriage_return"
$ WRITE FDLIN "        FORMAT                  fixed"
$ WRITE FDLIN "        SIZE                    80"
$ CLOSE FDLIN
$ CONVERT/PAD=%040/FDL=FIX GENRPD.DAT GENRPD.DAT
$!
$!
$! now run genrpd
$!
$ASSIGN/USER_MODE SYS$COMMAND SYS$INPUT
$ASSIGN 'LUW'.RPD SYS$OUTPUT
$RUNGENRPD
$DEASSIGN SYS$OUTPUT
$ DELETE GENRPD.DAT;*, FIX.FDL;*
$!
$!  now get the needed information to compile the rp-main(s)
$!
$ASSIGN 'LUW'.RDCOMP SYS$OUTPUT
```

```
$GENRPDFLAG = 0
$OPEN/READ  EFLE 'LUW'.RPD
$  RDLOOP:
$READ/END_OF_FILE=ENDMAIN EFLE EREC
$  LENG = 'F$LENGTH(EREC)'
$  DBMS = 'F$LOCATE("FOR DBMS", EREC)'
$  UN   = 'F$LOCATE("STORED ON", EREC)'
$  MN   = 'F$LOCATE("MODULE", EREC)'
$  DB   = 'F$LOCATE("DATA BASE", EREC)'
$  RM   = 'F$LOCATE("REMOTE/", EREC)'
$  HST  = 'F$LOCATE("RUN AT", EREC)'
$ IF 'MN'   .NES. 'LENG' THEN GOTO SAVMODNM
$ IF 'DB'   .NES. 'LENG' THEN GOTO SAVDBN
$ IF 'RM'   .NES. 'LENG' THEN GOTO RMLC
$ IF 'HST'  .NES. 'LENG' THEN GOTO SAVEHST
$ IF 'DBMS' .NES. 'LENG' THEN GOTO SAVEDBMS
$ IF 'UN'   .EQS. 'LENG' THEN GOTO RDLOOP
$GENRPDFLAG = 1
$UN1 = 'UN' + 16
$UNEND = 'F$LOCATE(".",EREC) - UN1
$PL := 'F$EXTRACT(UN1, UNEND, EREC)'
$IF DBMSNM .EQS. "ORACLE" THEN GOTO MAINPCC
$WS "A NEW DBMS TYPE MUST BE ADDED TO THE MAIN COMPILE PART OF THIS PROC
$GOTO EXIT
$!
$!  get the rp-main mod name
$!
$  SAVMODNM:
$MN1    = 'MN' + 7
$RPMN1 := 'F$EXTRACT(MN1, 10, EREC)'
$LENG   = 'F$LENGTH(RPMN1)'
$MN2    = 'F$LOCATE("ZZZ", RPMN1)
$IF 'MN2' .EQS. 'LENG'  THEN MODLOC = 0
$IF 'MN2' .NES. 'LENG'  THEN MODLOC = 2
$RPMN  := 'F$EXTRACT(MODLOC, 5, RPMN1)
$GOTO RDLOOP
$!
$!  get the remote/local status
$!
$  RMLC:
$RM1 = 'RM' + 13
$RMSW := 'F$EXTRACT(RM1, 1, EREC)'
$GOTO RDLOOP
$!
$!  get the database name
$!
$  SAVDBN:
$DB1 = 'DB' + 10
$DBN := 'F$EXTRACT(DB1, 30, EREC)'
$GOTO RDLOOP
$!
$!  get the host name
$!
$  SAVEHST:
$HST1 = 'HST' + 7
$HSTNM := 'F$EXTRACT(HST1, 3, EREC)'
$GOTO RDLOOP
$!
$!  get the dbms name
```

```
$!
$  SAVEDBMS:
$DBMS1 = 'DBMS' + 9
$DBMSNM := 'F$EXTRACT(DBMS1, 30, EREC)
$GOTO RDLOOP
$!
$!
$!  oracle precompile the rp-main (if needed)
$!
$  MAINPCC:
$PCC INAME='PL'.TMP LNAME='PL'.ERR USERID='CDM' -
 ONAME='PL'.COB INCLUDE=SYS$ORACLE: HOST=COB74 MAXLITERAL=160 REBIND=YES
$ON ERROR THEN WS "ORACLE ERROR IN RP-MAIN ''PL'.TMP"
$ON ERROR THEN GOTO EXIT
$COBOL/ANSI_FORMAT/CHECK=ALL/COPY_LIST/CROSS_REFERENCE/OBJECT='PL'.OBJ -
  /FIPS=74/NOLIST/CHECK=ALL/STANDARD=(SYNTAX)/DEBUG=ALL 'PL'.COB
$ON ERROR THEN WS "COBOL ERROR IN RP-MAIN ''PL'.TMP"
$ON ERROR THEN GOTO EXIT
$DELETE 'PL'.COB;*
$DELETE 'PL'.ERR;*
$!
$!  Compile the rpmain.c
$!
$vcc/debug/NOLIST/show=(include)/standard=portable -
     /noopt/OBJECT=CDMDIR:[TOOLS.DDLT]RPMAIN.OBJ/DEFINE=VAX RPMAIN.C
$delete rpmain.c;*
$GOTO RDLOOP
$!
$!  done generating rp-main
$!
$  ENDMAIN:
$CLOSE EFLE
$DEASSIGN SYS$OUTPUT
$IF GENRPDFLAG .EQ. 0 THEN GOTO MAINERR
$WS " "
$WS  "GENERATION OF REQUEST PROCESSOR MAIN COMPLETE"
$GOTO STARTLINK
$!
$!  there was an error in generating the rp-main
$!
$  MAINERR:
$WS "THE GENRPD HAD ERRORS.  EXAMINE ''LUW'.RPD"
$GOTO EXIT
$STARTLINK:
$DELETE 'LUW'.RPD;*
$DELETE 'LUW'.RDCOMP;*
$DELETE CMDIR:[RUNAREA]TOTTRANZ.EXE;*
$DELETE CMDIR:[RUNAREA]DB2TRANZ.EXE;*
$WRITE SYS$OUTPUT " - LINKING TOTTRANZ.EXE "
$WRITE SYS$OUTPUT " - LINKING DB2TRANZ.EXE "
$ASSIGN CDMTRAN.LINK SYS$OUTPUT
$!
$! inquire p5 "ENTER TWO LETTER NTM DIRECTORY PREFIX "
$! inquire p6 "ENTER NTM CLUSTER FOR THIS RP (T1V OR UIV) "
$!
$P5 = "GR"
$P6 = "UIV"
$P1 = "DB2TRAN"
$!
```

```
$!  NTMTAB.COM
$!
$!      15-APR-87
$!      M. DENMAN
$!
$!    UPDATE NTM TABLES APITBL.DAT, APTTBL.DAT
$!
$!
$ FLAG=0
$ P7=P5+P1
$ OPEN/READ APITBL.DAT CMDIR:[RUNAREA]APITBL.DAT
$ RD1:
$ READ/END_OF_FILE=CHK APITBL.DAT ENTRY
$ RPND=F$EXTRACT(0,9,ENTRY)
$ IF RPND .NES. P7 THEN GOTO RD1
$ FLAG=1
$ CHK:
$ CLOSE APITBL.DAT
$ IF FLAG .NES. 0 THEN GOTO NOUPD
$ GOTO UPD
$ NOUPD:
$ WRITE SYS$OUTPUT "  "
$ WRITE SYS$OUTPUT "RP MAIN ALREADY IN NTM TABLES"
$ WRITE SYS$OUTPUT "  "
$ GOTO LINKTRAN
$ UPD:
$ RPAPI=P7+"Z"+P6+"1"
$ RPAPT=P1+"Z9999010120001130N0"
$ OPEN/APPEND APITBL.DAT CMDIR:[RUNAREA]APITBL.DAT
$ OPEN/APPEND APTTBL.DAT CMDIR:[RUNAREA]APTTBL.DAT
$ WRITE APITBL.DAT RPAPI
$ WRITE APTTBL.DAT RPAPT
$ CLOSE APITBL.DAT
$ CLOSE APTTBL.DAT
$ P1= "TOTTRAN"
$ P7=P5+P1
$ RPAPI=P7+"Z"+P6+"1"
$ RPAPT=P1+"Z9999010120001130N0"
$ OPEN/APPEND APITBL.DAT CMDIR:[RUNAREA]APITBL.DAT
$ OPEN/APPEND APTTBL.DAT CMDIR:[RUNAREA]APTTBL.DAT
$ WRITE APITBL.DAT RPAPI
$ WRITE APTTBL.DAT RPAPT
$ CLOSE APITBL.DAT
$ CLOSE APTTBL.DAT
$ WRITE SYS$OUTPUT "  "
$ WRITE SYS$OUTPUT "TWO NTM TABLES UPDATED WITH RP "
$ WRITE SYS$OUTPUT "  "
$!
$LINKTRAN:
$@SYS$ORACLE:LFOR CMDIR:[RUNAREA]TOTTRANZ -
CDMDIR:[TOOLS.DDLT]TOTTRAN.OBJ,-
CDMDIR:[TOOLS.DDLT]RPMAIN.OBJ,-
'PL'.OBJ,-
CDMDIR:[COM]CDMI/OPTIONS,-
SYS$ORACLE:SQLLIB/LIB,-
ui( ir:[fe]feolb/inc=(fluierr),-
CDMDIR:[COM]CDMUI.OPT/OPT,-
CDMDIR:[COM]CDMNTM.OPT/OPTIONS N
$@SYS$ORACLE:LFOR CMDIR:[RUNAREA]DB2TRANZ -
```

```
CDMDIR:[TOOLS.DDLT]DB2TRAN.OBJ,-
CDMDIR:[TOOLS.DDLT]RPMAIN.OBJ,-
'PL'.OBJ,-
CDMDIR:[COM]CDMI/OPTIONS,-
SYS$ORACLE:SQLLIB/LIB,-
uidir:[fe]feolb/inc=(fluierr),-
CDMDIR:[COM]CDMUI.OPT/OPT,-
CDMDIR:[COM]CDMNTM.OPT/OPTIONS N
$DEASSIGN SYS$OUTPUT
$DELETE CDMTRAN.LINK;*, RPMAIN.OBJ;*, 'PL'.OBJ;*
$WRITE SYS$OUTPUT "LINKING COMPLETED"
$DEFINE/NOLOG IISSGLIB "CDMDIR:[TEST]GENOLB.OLB"
$DEASSIGN TOOLOLB
$!
```

☆U.S. GOVERNMENT PRINTING OFFICE: 1992 - 648-127/62343